Universidad Autónoma de Madrid
Escuela Politécnica Superior

Proyecto fin de grado

DETECCIÓN DE TENDENCIAS EN TWITTER UTILIZANDO MINERÍA DE DATOS ADAPTATIVA

Grado en Ingeniería Informática

Natalia Roales González
Julio 2014

DETECCIÓN DE TENDENCIAS EN TWITTER UTILIZANDO MINERÍA DE DATOS ADAPTATIVA

AUTOR: Natalia Roales González
TUTOR: Gema Bello Orgaz
Co-Tutor: David Camacho Fernández

Grupo de la EPS: AIDA
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2014

Resumen

Resumen

Actualmente, las redes sociales se han convertido en una importante fuente de comunicación a nivel mundial. La gran cantidad de interacciones que se producen entre sus usuarios proporcionan una inmensa cantidad de información sobre sus preferencias, lo cual resulta verdaderamente útil para la detección de tendencias. Una de las redes sociales más relevantes y populares es Twitter, donde millones de usuarios comparten sus comentarios y opiniones a través de mensajes de texto cortos. Estos mensajes pueden ser extraídos y a continuación tratados mediante diferentes técnicas de minería de datos, para así poder distinguir grupos de tendencias de opiniones sobre un tema.

Este trabajo tiene como objetivo el estudio de este fenómeno, centrándose en un tema en concreto: Eurovisión. Este festival es un referente en el mundo de la canción, pero también puede ser observado desde un ámbito geográfico, político, artístico, histórico y social. Se tratarán de detectar conexiones entre distintos países, intentando predecir a través de las mismas un ranking de votaciones. Posteriormente, utilizándose las mismas técnicas de minería de datos, el trabajo podría extrapolarse a otros campos.

Palabras Clave

Twitter, Minería de datos, Eurovisión, Ranking, Detección de tendencias.

Abstract

Nowadays, social networks have become in a important source of mundial comunication. The large volume of interaction that are making between different users provides a big amount of information about their preferences, this turn out to be very useful for the trend detection. One of the mot popular social network is Twitter, where million of users share their commments and opinion through short messages. These messajes can be extrated and lately be processed through different techniques of data mining, to lately be able to make out different trend groups of opinion about a topic.

This essay has as a objective the study of this phenomenon, focus on a specific topic: Eurovision. This festival is a reference in the music world, but it can also be seen from a geographic, political, artistic, historical and social context.This will try to detect conection between diferent countries, trying to predice thourgh the last ones a ranking of votes. . Afterwords, using these tecnquies of mining datum, this could be extrapolate to other fields.

Key words

Twitter, Data mining, Eurovision, Ranking, Trend detection.

Agradecimientos

En primer lugar, gracias a mi tutora Gema, por prestarme toda su ayuda, por su tiempo y dedicación, y por haberme animado a hacer este trabajo. A mis compañeros de laboratorio, quienes han estado dispuestos a echarme una mano siempre y han hecho que las horas de *TFG* sean mucho más amenas.

A mis compañeros de carrera, por hacer que estudiar Informática haya sido algo tan divertido. Gracias a todos aquellos que en todo momento me han echado un cable, y a quienes han decidido acompañarme durante esta etapa tan importante para mí.

A mis profesores, quienes me han enseñado tantísimas cosas, algunos más que otros, sacrificando en muchas ocasiones su tiempo para resolver mis problemas o para ayudarme con sus tutorías.

A Elena y Ainhoa, mis compañeras de batallas. Elena, gracias por tus hojas amarillas y por ayudarme con tus super-horarios con descansos. Ainhoa, gracias por esos ratos de biblioteca tan productivos en los que tantas dudas hemos resuelto antes de cualquier examen. Chicas, gracias por vuestra inestimable ayuda y por estar ahí siempre que lo he necesitado.

A Santi, por haber colaborado conmigo al desarrollar la base de la que parte este proyecto, siendo un excelente compañero. Gracias también por haberme ayudado a dar mis primeros pasos en el mundo de LaTeX.

A todos mis amigos, quienes logran conseguir que me olvide por un rato de mis preocupaciones cuando las prácticas o exámenes me ahogan.

A ti, Álvaro, simplemente por estar siempre ahí a mi lado. Gracias por tu apoyo incondicional, por aconsejarme como tú solo sabes, y por conseguir que sonría incluso cuando todo parece que se tuerce. Gracias por sacar lo mejor de mí.

Finalmente, quiero dar las gracias sobre todo a mi familia, en especial a mis padres y a mi hermano. Gracias por vuestro apoyo, por vuestro cariño, por creer siempre en mí, y por ayudarme a conseguir todo lo que me proponga. Sin vosotros no hubiese sido posible nada de esto. Gracias por todo.

Natalia Roales González
Julio de 2014

Índice general

Índice de figuras

Índice de tablas

1 Introducción

La minería de datos puede definirse inicialmente como un proceso de descubrimiento de nuevas y significativas relaciones, patrones y tendencias al examinar grandes cantidades de datos [1]. Cuando disponemos de abundantes volúmenes de información y necesitamos analizarlos de manera efectiva, ya sea para comprenderlos mejor o para predecir ciertos comportamientos, necesitamos ayuda computacional.

Unas de las mayores fuentes de datos son las redes sociales digitales. Este servicio permite a los usuarios conectarse con otros, ya sea por una relación de amistad o de intereses comunes, que dependerán del tipo de contenido de la red. Cada usuario crea su propio perfil en la red, y comparte información con otras personas — de manera pública o privada, según el nivel de privacidad — de manera instantánea, por lo que la cantidad de datos que se generan es inmensa.

Algunas de las redes sociales más relevantes y concurridas de Internet son Twitter, Facebook y Youtube. Twitter es uno de los servicios de microblogging más populares en la Web. Permite a sus usuarios publicar tweets, que son mensajes cortos de texto plano con una longitud de hasta 140 caracteres. Este límite imita el sistema de mensajería SMS de los teléfonos móviles. Los usuarios registrados, llamados twitteros, pueden seguir a otros con el fin de recibir las publicaciones que estos realicen. Además, un usuario puede utilizar hashtags, mencionar a otros usuarios, o retwittear los mensajes que le parezcan interesantes. En Twitter, los mensajes son públicos por defecto, por lo que los datos son totalmente analizables. Además, cada tweet suele contener opiniones personales de los usuarios, lo cual es bastante interesante en el análisis de datos.

Este trabajo se basa en la explotación y el análisis de la información disponible en esta gran fuente de datos, Twitter, mediante la aplicación de técnicas de minería de datos. Existen infinitos temas sobre los que la gente opina en esta red social, y por ello debemos acotar nuestro proyecto a un tema en concreto. El Festival de Eurovisión, uno de los eventos musicales más importantes en Europa, es un buen tema para desarrollar este proyecto, siendo desde hace años objeto de investigación en distintos campos del conocimiento. Su sistema de votaciones aporta información muy valiosa sobre los países que participan en él, aunque a simple vista es prácticamente imposible obtener ninguna conclusión. Por esta razón, es interesante la aplicación de la minería de datos en este caso.

El presente documento refleja todo el trabajo llevado a cabo. En este capítulo, se presentan la motivación y los objetivos marcados en el proyecto. En el Capítulo 2, se habla del estado del arte. En el Capítulo 3, se describe al detalle la arquitectura de la aplicación desarrollada. En el Capítulo 4, se exponen la experimentación y el análisis de los resultados. Por último, en el Capítulo 5 se finaliza con una serie de conclusiones, y se plantea el trabajo futuro a realizar.

1.1. Motivación del proyecto seleccionado

En los últimos años, las redes sociales se han convertido en un importante medio de comunicación. Hoy en día casi todo el mundo que disponga de conexión a Internet tiene un perfil en alguna de ellas, y mucha gente las usa activamente, compartiendo un flujo de información ingente que crece cada día de manera exponencial. Pero además de darle un uso tan cotidiano a este servicio, como puede ser el hecho de contactar con nuestros amigos o compartir información con el resto del mundo, podemos ir más lejos y sacarle partido a todos esos datos. Gracias a potentes técnicas de minería de datos podremos darles un uso mejor, extrayendo conocimiento de manera automática o realizando predicciones sobre comportamientos o hechos futuros.

1.2. Objetivos

Los objetivos de este proyecto se describen a continuación:

1. **Selección de las herramientas de trabajo**: En primer lugar, se debe realizar un estudio para la selección de las herramientas con las que vamos a trabajar, ya sean lenguajes de programación, APIs de consulta y descarga de datos, o gestores de bases de datos. Se llevará a cabo una comparativa para analizarlas y destacar las más apropiadas a la hora de realizar cada una de las tareas del proyecto.

2. **Extracción de tweets**: Tras haber elegido una librería que trabaje con el API de Twitter, se procederá a extraer y almacenar los tweets. Previamente, se debe establecer el ámbito de las búsquedas en Twitter y, además, se debe realizar un pequeño estudio sobre qué campos son o no relevantes.

3. **Explotación de datos**: El siguiente paso ha sido crear un índice para permitir realizar búsquedas sobre el texto del material extraído, siendo cada documento del índice cada uno de los tweets. Se implementarán todos los métodos o funciones necesarias para poder explotar la información indexada.

4. **Generación de rankings**: Trataremos posteriormente todos los datos de los cuales disponemos, agrupándolos y formando rankings de carácter geográfico. Se consultarán en el sitio web de Eurovisión los resultados de las votaciones emitidas en la última edición del Festival.

5. **Análisis y conclusiones**: Se analizarán los rankings generados por país, comparándose con los resultados oficiales del Festival. Finalmente, y como resultado de todo el trabajo realizado, se extraerán algunas conclusiones acerca de los análisis realizados.

2 Estado del arte

En este capítulo se introduce el estado del arte, presentándose estudios ya realizados y relacionados con este proyecto, ambos enmarcados en los campos de minería de datos, redes sociales e investigación sobre Eurovisión.

2.1. Twitter y la minería de datos

Al igual que muchas otras redes sociales, Twitter es una gran fuente de datos. Por ello, gran multitud de estudios enmarcados en el campo de la minería de datos se han centrado en esta plataforma social.

Sin duda, estudios basados en marketing destacan sobre muchos otros trabajos y proyectos desarrollados en torno a Twitter [2]. Las marcas aprovechan toda la información que los usuarios generan al comunicarse con familiares y amigos, explotándola y descubriendo nuestros intereses, además de conocer qué se está comentando sobre una marca en concreto. También envían spam a los usuarios de Twitter, y este estudio recoge que esta red social es muy susceptible a ese tipo de prácticas. Se analizan marcas de distintos sectores: marcas del sector textil, de la industria automovilística, del sector de hostelero, etc. Además, se trata de detectar qué porcentaje de los tweets corresponden a opiniones acerca de las marcas, y se obtiene que un 20 % de las menciones sobre una marca pertenecen a esa categoría.

Recientemente han aparecido varios trabajos que muestran el potencial de Twitter para poder hacer seguimiento y predicción de brotes de enfermedades. Los usuarios de Twitter pueden publicar comentarios acerca de enfermedades y sus síntomas, y realizando una geolocalización de los mismos se podrían utilizar para detectar zonas de posibles brotes de enfermedades.

Basándose en esta idea, la universidad de Northwestern ha desarrollado un sistema para detectar brotes de gripe [3]. El sistema[1] extrae continuamente textos relacionados con enfermedades y utiliza técnicas de minería de datos basados en texto para mostrar gráficas y mapas de la enfermedad.

Por último, también se han llevado a cabo trabajos que estudian el impacto de Twitter en otros medios de comunicación, en concreto, la televisión [4]. Este estudio se centró en el análisis de las audiencias, poniendo de ejemplo el festival de Eurovisión, aunque no solo tuvo en cuenta la comunidad europea para desarrollar este trabajo. Una cadena televisiva

[1] http://pulse.eecs.northwestern.edu/ kml649/flu/

en Australia retransmite este evento con cierto retraso, por lo que anima a los televidentes a seguir el concurso de forma más dinámica e interactiva a través de hashtags en Twitter. Se muestra cómo las cadenas televisivas utilizan este sistema para conocer sus audiencias. Esto además nos parece interesante, ya que Eurovisión va a ser también nuestro objetivo al investigar.

2.2. Eurovisión como objeto de investigación

El festival de Eurovisión es un referente en el mundo de la canción, en el que los países de la Unión Europea participan con una actuación musical, mostrando al resto sus tendencias musicales. Su dinámica ha ido evolucionando a lo largo de muchos años, variando el sistema de votaciones usado en el concurso.

En 1975 se implantó el sistema por el cual los países votan con 12 puntos a su canción favorita, 10 a la segunda, y sucesivamente desde 8 puntos hasta 1 [5]. Las votaciones de cada país son emitidas por un jurado. En 1997, algunos países ponen en práctica el sistema del televoto, y en 1998 todos los países tienen la oportunidad de utilizarlo, aunque seguía existiendo un jurado de reserva en cada país por si el televoto no funcionase. Este sistema permitía al público del festival votar por su canción favorita a través de la línea telefónica.

Desde 2001, todos los países están obligados a utilizar televoto, excluyendo a aquellos en los que por problemas técnicos no fuese posible, y a partir del 2009 el 50 % de los votos en cada país es dado por los televotos, siendo el otro 50 % otorgado por un jurado profesional. Este sistema se llevó a cabo para atenuar el efecto diáspora, o lo que es lo mismo, para no favorecer a los países inmigrantes.

Por último, en el 2013 se creó una aplicación móvil[2] desde donde poder emitir votos. Gracias a este sistema combinado, las votaciones aportan información muy valiosa que puede ser extrapolable a tendencias políticas y sociales, y por ello numerosos estudios pertenecientes a diferentes áreas de investigación lo toman como objetivo.

El conjunto de datos de tipo social que se pueden obtener a partir del festival, sus votaciones, y la interacción entre los diferentes países ha sido estudiado desde los años 90 utilizando diversas técnicas de minería de datos.

Uno de estos primeros trabajos analizó cómo las estructuras políticas y culturales que forman la comunidades se ven, de la misma forma que en los estudios anteriores, reflejadas en el festival de Eurovisión [6]. Como resultado de esta investigación se encontraron tres agrupaciones, cada una de ellas unida por diferentes intereses y sentimientos, destacando el bloque occidental, el norte y el mediterráneo. Mientras que el bloque occidental se puede ver como una coalición basada en intereses históricos y políticos, el bloque norte está definido por los códigos de idiomas y culturales comunes. En cambio, el bloque mediterráneo es mucho más difuso que los dos anteriores, y logra su alianza gracias a las experiencias culturales compartidas.

Otro trabajo posterior al inicio del televoto ofrece un análisis sobre el comportamiento de las masas, evaluando el impacto del sistema del televoto, y contrastando los esquemas de votación utilizados durante años con los resultados del concurso en cada ocasión [7].

[2] http://www.eurovision.tv/static/app/

En este artículo se realiza también un estudio sobre la estructura de las comunidades detectadas, agrupadas mediante algoritmos de clustering, y cómo ha evolucionado a lo largo de la historia del festival.

Los resultados de este estudio muestran como, a través del televoto, pueden ser identificadas comunidades estables a través del tiempo, ya sea un grupo de países con un pasado en común o con raíces históricas y culturales similares. Además, este sistema resalta las evidencias del efecto diáspora que se ha mencionado anteriormente.

A través del televoto también, se pretende localizar una serie de agentes que influyan en la conexión y afinidad entre países [8]. En ambos trabajos [7][8] se han utilizado algoritmos de clustering, en este caso basándose en las similitudes entre el número de puntos que los países asignan a otros participantes. Los resultados de este estudio muestran que no es la cercanía geográfica la que agrupa a las comunidades entre países, sino que los enlaces creados por los grupos políticos de cada nación o sus gobiernos son aún más fuertes. Un dato curioso es que Reino Unido es mucho más afín los países europeos que, por ejemplo, Francia.

3 Arquitectura de la aplicación

Antes de describir en detalle cada una de las fases de este proyecto, es conveniente dar un vistazo general. Para ello, en este capítulo se esboza un esquema simple de la arquitectura de la aplicación diseñada e implementada, y se realiza un breve resumen de su funcionamiento.

3.1. Esquema de la arquitectura

El esquema de la arquitectura de la aplicación desarrollada en este proyecto se representa mediante la Figura 3.1.

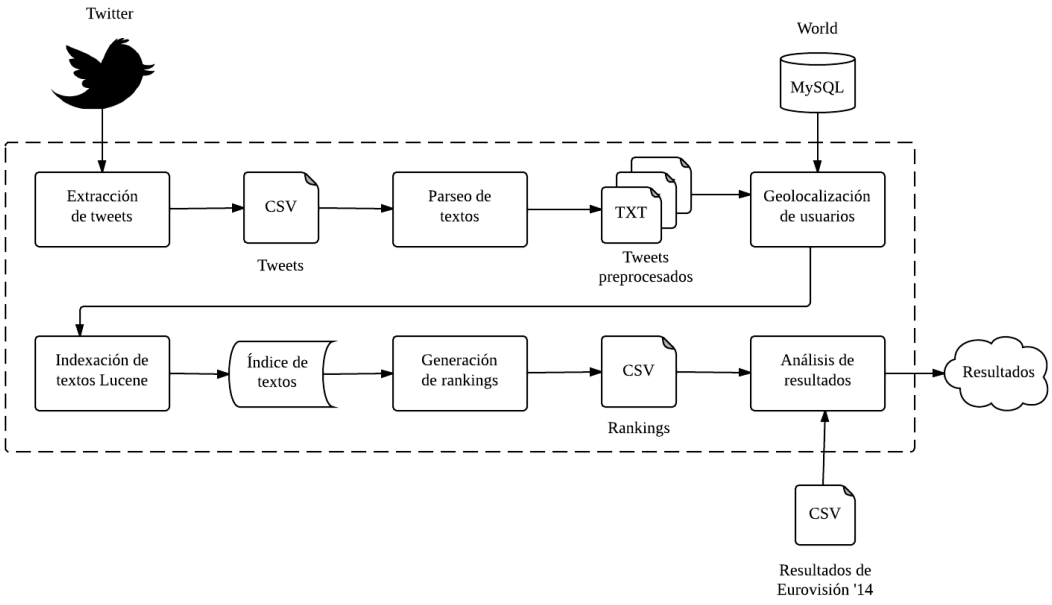

Figura 3.1: Esquema de la arquitectura de la aplicación

El área encerrada entre líneas discontinuas representa la aplicación con cada uno de sus módulos y elementos generados internamente, las flechas dirigidas indican el flujo de datos, y los elementos externos son las entradas y la salida. A continuación, se da una breve descripción del funcionamiento de este esquema.

3.2. Síntesis del funcionamiento de la aplicación

El funcionamiento de la aplicación puede sintetizarse de la siguiente manera:

1. En primer lugar, la aplicación se conecta a Twitter por medio de su API, extrayendo los tweets y almacenándolos en un fichero CSV. Para trabajar con el API se invoca a la librería Twitter4J.
2. Luego, se preprocesan los datos extraídos, transformando los ficheros CSV resultantes en ficheros de texto individuales, cada uno almacenando un tweet.
3. En algún caso, los campos geográficos del fichero se encontrarán vacíos. Debe deducirse entonces la información que falta, ya que nuestro objetivo es la generación de rankings por países, y para ello la aplicación se conecta a la base de datos World. A partir del lugar de procedencia del autor de cada tweet — contenido cada mensaje en un fichero distinto — se obtendrá el país de geolocalización.
4. Tras haber geolocalizado los mensajes que lo necesitasen se procede a indexar todos los ficheros de texto, generándose un índice que se almacena en disco para su posterior acceso.
5. Después, mediante diversas búsquedas sobre el índice creado, se generarán rankings por países. Estos rankings serán almacenados en ficheros CSV, y pretenderán predecir las puntuaciones del festival.
6. Por último, enfrentando los rankings generados a las puntuaciones reales del Festival, la aplicación analizará los resultados obtenidos en este proyecto.

En las próximas secciones, se describirán en detalle tanto la implementación de la arquitectura como cada una de las fases del procedimiento que realiza la aplicación.

3.3. Extracción de datos en Twitter

En esta sección se detalla paso a paso todo el procedimiento de extracción de datos en Twitter, cuyas fases son el registro en la plataforma para desarrolladores, la búsqueda y descarga de los tweets.

3.3.1. Registro en la plataforma para desarrolladores

El primer paso en nuestro trabajo, previo a la extracción de tweets, es el registro en la plataforma para desarrolladores[1] de Twitter. Esta plataforma permite a un usuario registrar su aplicación en desarrollo y generar una serie de claves, que utilizará posteriormente para autenticarse desde la API de Twitter seleccionada.

Las claves que la plataforma proporciona a cada desarrollador son cuatro: API key o Consumer key, API secret o Consumer secret, Access token, y Access token secret. La

[1] https://dev.twitter.com/

autenticación desde la API se realizará de manera segura a través del protocolo OAuth, permitiendo a los usuarios comunicarse con Twitter sin compartir toda su identidad.

Para registrarse como desarrollador, es necesario haberse registrado anteriormente en Twitter. Una vez hecho esto, los únicos datos que la plataforma solicita acerca de la aplicación son su nombre, descripción y URL de acceso. Por último, es posible limitar los permisos de la aplicación; nuestra aplicación únicamente dispondrá de permisos de lectura.

3.3.2. Extracción de tweets

Tras habernos registrado como desarrolladores, ya podemos comenzar a extraer mensajes de Twitter. Debemos plantearnos, antes de nada, qué queremos extraer y cómo. O lo que es lo mismo, cuáles van a ser los criterios de extracción y qué API vamos a utilizar. Se detalla, a continuación, la respuesta a cada una de estas dos cuestiones.

Criterios de búsqueda

No todos los países europeos participan en Eurovisión. Sin embargo, miles de personas alrededor de todo el mundo opinan sobre el Festival, dejando constancia en Twitter. Por ello, la cantidad de material que podemos encontrar en la red social sobre Eurovisión es de dimensiones astronómicas.

Debemos tener muy claro cuál va a ser nuestro objeto de búsqueda, contemplando algún criterio de consulta que nos permita acceder a un gran número de mensajes, escritos por una multitud de usuarios lo más amplia y diversa posible.

Para escribir en Twitter acerca de cualquier tema, los usuarios utilizan hashtags. De manera que, si realizamos una consulta sobre un determinado hashtag, podremos leer todo lo que se ha dicho sobre el tema que le corresponda. Por esta razón, se ha seleccionado un hashtag como criterio de búsqueda para la extracción de datos, y ese hashtag será *#Eurovision*.

Coincide con el nombre del festival, así que cualquier usuario que quiera introducir un hashtag en su publicación, independientemente de su origen, lo utilizará.

Selección de librerías para el manejo de la API de Twitter

Ha sido seleccionada, para la extracción de los tweets, la librería Twitter4J, implementada sobre Java. La razón por la cual se ha elegido dicha librería, han sido las habilidades y conocimientos sobre programación adquiridos a lo largo de la carrera; mi destreza en el uso de Java es mucho mayor frente a cualquier otro lenguaje.

Delimitación de los campos a filtrar sobre cada tweet

Existe, en la librería Twitter4J, una clase llamada Status. Esta clase contiene en sus atributos toda la información correspondiente a un determinado tweet.

Los atributos — o campos — de la clase Status son muy completos, pero algunos de ellos son realmente importantes dentro del ámbito de nuestro trabajo. Por ello, descartaremos el resto de atributos que no necesitemos.

En la Tabla 3.1 se pueden ver los campos de cada tweet que nosotros vamos a necesitar para llevar a cabo nuestro proyecto:

Atributo	**Descripción**
ID	Identificador del mensaje
Iso Language Code	Código internacional del idioma en el que se escribió el mensaje
Created At	Fecha de publicación del mensaje
Place Country	País desde donde se publica el mensaje, localizado a través de coordenadas GPS
Place Country Code	Código internacional del país de geolocalización
User Screen Name	Nick del usuario que ha publicado el mensaje
User Location	Procedencia del usuario según su perfil
Is Retweet	Valor booleano que indica si el mensaje es o no un retweet
Retweet Count	Número de veces que el mensaje ha sido retweeteado. En caso de ser un retweet, este contador será nulo
Favorite Count	Número de veces que los usuarios han marcado este mensaje como favorito
Text	Mensaje de texto o cuerpo del tweet

Tabla 3.1: Atributos más relevantes de la clase Status

Implementación de una aplicación para el manejo de la librería

Para facilitar el manejo de la librería Twitter4J, y con el fin de extraer los datos de Twitter, ha sido implementada una clase java llamada ExtraeTweets. Esta clase consta de un constructor sin argumentos y dos métodos, detallados en esta sección. Existe en ella además un atributo privado de la clase Twitter, cuya funcionalidad será el acceso a la API.

- **Constructor:**

  El constructor instancia la clase, autenticando al usuario a través del protocolo de seguridad e inicializando un objeto de tipo Twitter, que será el atributo de la clase.

- **Método *buscarTweets*:**

  El método *buscarTweets* realiza una búsqueda en Twitter, retornando en una lista el resultado de la misma. Recibe como parámetros la cadena de texto a buscar y el número de tweets que se desean extraer.

  El procedimiento que sigue este método, es el siguiente:

  1. Creamos un objeto de tipo Query a partir de la cadena de texto que queremos buscar. Este objeto será la instancia de la consulta que enviaremos a Twitter.
  2. Indicamos sobre la consulta (sobre el objeto de tipo Query) la fecha límite de los tweets que serán extraídos.

3. Especificamos sobre la consulta el número de tweets que queremos extraer.
4. Mientras no alcancemos el número de tweets deseado...
   - Extraemos 100 tweets, o lo que quede hasta llegar a la cantidad solicitada.
   - Insertamos los nuevos mensajes en una lista junto con los demás.
   - Almacenamos el identificador del último tweet extraído, ya que en la próxima iteración lo necesitaremos para no volver a extraer los mismos mensajes de nuevo.
   - Cada vez que el tamaño de la lista llegue a un múltiplo de 675, imprimimos los tweets en un fichero y esperamos un minuto antes de continuar para no sobrepasar el límite de extracción de datos a través de la API.
   - Si al final la iteración hemos conseguido el número de tweets deseado, retornamos la lista. Si no, realizamos una nueva iteración.

La implementación de esta función está pensada de manera que nunca llegue a alcanzarse el límite de extracción de datos a través de la API. Pero sin en algún momento, por razones desconocidas, llega a alcanzarse dicho límite, el uso de la API queda restringido.

En ese momento, el método imprimirá en un fichero los tweets almacenados en la lista, y será pausado 5 minutos, aunque el tiempo de espera que establece la API es de 15 minutos.

- **Método *imprimeTweetsFichero*:**

  El método *imprimeTweetsFichero*, imprime en un fichero CSV una lista de tweets, incluyendo uno en cada fila. Recibe como parámetros de entrada una lista de objetos de la clase Status y una ruta donde crear el nuevo fichero. Su retorno es vacío.

  El funcionamiento de este método es muy simple; en primer lugar, se imprime la cabecera del fichero, que será el nombre de los campos de cada tweet. Posteriormente, se imprimen los campos de cada uno de los tweets, todos en la misma fila. La separación entre campos, al igual que la separación de los nombres en la cabecera, se determina con punto y coma. La separación entre los tweets se realiza mediante saltos de línea.

Limitaciones de la API y de la librería utilizada

Twitter impone una serie de limitaciones a los desarrolladores en el uso de su API, que consisten en la restricción de la consulta y descarga de datos. Esto implica cierto grado de dificultad a la hora de desempeñar nuestro trabajo de extracción de tweets, ya que dichas limitaciones nos afectan directamente.

La limitación más restrictiva de todas ha sido la incapacidad de extracción de mensajes con más de una semana de antigüedad, al realizar cualquier consulta a través de la API. Ya que el objetivo en el cual se resume este proyecto se basa en la predicción de puntuaciones, el tiempo corre en nuestra contra, pues recopilar mensajes posteriores a la fecha del Festival es totalmente inútil.

Otra de las restricciones a destacar, es el límite de consultas que un usuario puede formular en un minuto. El valor máximo de consultas que Twitter establece puede variar,

por lo que conviene ser precavido y moderar el flujo de peticiones. Si el usuario alcanza este límite, deberá esperar 15 minutos hasta realizar la siguiente petición. Para evitar esto, la estrategia seguida ha consistido en pausar durante un minuto nuestro proceso cada vez que se extrajesen 675 tweets.

Nos hemos encontrado también con dos limitaciones más, a causa del uso de la librería Twitter4J. La primera de ellas, se impide visualizar los datos de todo aquel usuario que marca como favorito un tweet. Es decir, la librería nos permite obtener el número de veces que ha sido marcado un tweet como favorito, pero no por quién. Por último, no es posible obtener mediante una misma petición los datos de un retweet y de su tweet de procedencia. Deben realizarse dos peticiones distintas, lo que implica aumentar la tasa de peticiones por minuto, acelerando el alcance del límite de consultas por minuto.

3.4. Preprocesado de datos

Tras realizar todo el proceso de extracción de tweets, debemos implementar una solución eficiente para la búsqueda y acceso sobre los datos extraídos, ya que son muy numerosos y contienen mucha información que explotar. Para ello, es requisito fundamental preprocesar los datos, dando a cada mensaje un formato que facilite el manejo de todo el conjunto. Se realizará la escritura de los mensajes en ficheros de texto, almacenando cada tweet en un fichero distinto. A continuación, se describe el proceso de preprocesado de los datos.

3.4.1. Escritura de los mensajes en ficheros de texto

Previamente, cada lista de tweets extraída ha sido grabada en un fichero CSV, invocándose al método *imprimeTweetsFichero*. Después, se ha implementado la clase java *Parser*, cuya funcionalidad es la del parseo de información.

Dentro de esta clase, se ha implementado el método *CSVParser*. Este método realiza el trabajo de almacenar cada tweet en un fichero de texto, y recibe dos cadenas de texto como parámetros: la primera referencia a la ruta donde se encuentran los ficheros CSV donde se encuentran los mensajes extraídos de Twitter, y la segunda referencia a la ruta donde se guardarán los nuevos ficheros de texto.

El procedimiento que sigue este método consiste en leer todos los ficheros del directorio de entrada, línea a línea, creando un nuevo fichero de texto por cada línea, y grabándolo en el directorio de salida. A continuación, se especifica el formato de cada fichero de texto de salida.

Características de los ficheros de texto

Cada fichero contiene los campos de un tweet, delimitados por saltos de línea, y cada campo se encuentra precedido del carácter '#'. Un ejemplo de fichero de texto sería el siguiente:

```
#464510904047256000
# en
# Thu May 08 23:03:24 CEST 2014
#
#
# dogfish44
# Wigan, UK
# false
# 0
# 0
# Goodbye to Ireland, Georgia, Macedonia, Israel and Lithuania! #eurovision
```

Este fichero se llama 464510904047256000.txt, y corresponde al valor del ID del tweet seguido de su extensión. Observamos que, en este caso, el cuarto y el quinto campo están vacíos. Esta es la razón por la cual cada campo va precedido de un símbolo, puede existir algún tweet para el cual uno de los campos se encuentre vacío. De esta forma, tendremos todos los atributos perfectamente localizados.

3.4.2. Geolocalización de los tweets

Puede darse el caso en que los datos de geolocalización de un tweet no existan, porque el usuario que lo haya escrito no desee o no pueda añadir la localización a sus tweets vía GPS. Es decir, es posible que los atributos Place Country y Place Country Code sean nulos o cadenas de texto vacías. Son muy pocos los usuarios que geolocalizan sus publicaciones, por lo que debemos deducir el país de procedencia del tweet, ya que nuestro objetivo es generar un ranking agrupado por países.

Una opción es obtener el país de procedencia del usuario, accediendo a sus datos de su perfil. Existe un atributo llamado User Location, que corresponde a la ubicación del usuario según su perfil. Además, disponemos en Internet de numerosas bases de datos que almacenan los nombres de todos los lugares del mundo, ya sean países, ciudades, pueblos o regiones. Se hará uso de la base de datos World, buscando el país correspondiente a la ubicación del usuario en caso de ser necesario. Una vez conseguida esta información, el preprocesado de datos habrá finalizado satisfactoriamente.

Base de datos World

Se trata de una base de datos MySQL que contiene un registro de todos los lugares del mundo, escritos en inglés en caso de que tengan traducción a ese idioma. Las siete tablas que forman World se recogen en la Tabla 3.2.

Esta base de datos es muy completa, pero solamente necesitaremos un conjunto reducido de las tablas y sus atributos. En concreto, realizaremos la búsqueda de la ubicación de cada usuario sobre tres tablas: Countries, Cities y Regions. En el siguiente subapartado se desglosará el procedimiento llevado a cabo para realizar estas búsquedas.

Tabla	Descripción
Cities	Registra todas las ciudades y pueblos del mundo. Sus atributos más importantes son el nombre de la ciudad y el código ISO de país en el que se encuentra.
Citynames	Contiene los nombres de todas las ciudades almacenadas en la tabla cities, registrando además el código ISO del idioma en el que se escriben y una clave externa que apunta a la tabla cities.
Countries	Registra todos los países del mundo. Sus atributos más importantes son el nombre del país y su código ISO.
Countrynames	Contiene los nombres de todos los países almacenados en la tabla countries, registrando además el código ISO del idioma en el que se escriben y su código ISO.
Regions	Registra todas las regiones del mundo. Sus atributos más importantes son el nombre de la región y el código ISO de país en el que se encuentra.
Regionnames	Contiene los nombres de todas las regiones almacenadas en la tabla regions, registrando además el código ISO del idioma en el que se escriben, y una clave externa que apunta a la tabla regions.
Locales	Almacena un registro de todos los idiomas del mundo. Sus atributos son el nombre del idioma y su código ISO.

Tabla 3.2: Tablas de la base de datos World

Tratamiento de la ubicación del usuario

El proceso de tratamiento de la ubicación de un usuario, desde que extraemos la información de su perfil hasta que conseguimos averiguar su país de procedencia, tiene tres fases. Estas fases se resumen en: 1. Fragmentación de la ubicación en varias subcadenas; 2. Limpieza de caracteres no válidos en las subcadenas; y 3. Búsqueda de las subcadenas en la base de datos World. A continuación, se describe la implementación realizada en cada una de las tres fases.

- **Fragmentación de la ubicación:**

  Es muy probable que un usuario escriba más de un lugar en la información de su ubicación; por ejemplo, alguien podría indicar que vive en *"Madrid, España"*. Es por eso por lo que debemos fragmentar la ubicación completa en varias subcadenas de texto. Se ha hecho una selección de los caracteres que los usuarios utilizan más frecuentemente para separar nombres, y los caracteres elegidos que marcarán la separación en subcadenas serán seis:
  «, – / | ( )» . Una vez fragmentada la cadena de texto original, todas las subcadenas se guardarán en una lista.

- **Limpieza de caracteres no válidos:**

  En segundo lugar, debemos limpiar las subcadenas de la lista obtenida en la fase de fragmentación. Los caracteres no alfabéticos, a excepción de los apóstrofos, deben ser eliminados. En este caso van a ser sustituidos por espacios, ya que el nombre de una ciudad o de un país puede estar compuesto por más de una palabra. Los espacios innecesarios serán eliminados, es decir, las cadenas de espacios se reducirán a un

solo espacio, y los espacios situados al principio o al final de cada cadena también serán suprimidos. Una vez hecho esto sobre todas las cadenas de la lista, podemos proceder a realizar las búsquedas en la base de datos.

- **Búsqueda en la base de datos:**

  El último paso, será buscar en la base de datos World los nombres almacenados en la lista, que corresponden a países, ciudades, pueblos y regiones. Como no sabemos en qué tabla hay que buscar cada nombre, se realizarán tres búsquedas similares, primero en la tabla Countries, después en la tabla Cities, y por último en la tabla Regions.

  En los tres casos, la consulta estará compuesta de una cláusula WHERE y de una cláusula EXISTS, que a su vez contendrá otra consulta anidada. En la cláusula WHERE compararemos los nombres almacenados en la lista con el atributo name de cada tabla, utilizando el operador lógico OR. Y, en la cláusula EXISTS, comprobamos que el código ISO del país corresponda al de los países participantes en Eurovisión. Para eso ha sido creada una tabla llamada Voters, que contiene el código ISO de todos los países votantes. También ha sido creada la tabla Singers, que contiene el código ISO de todos los países que cantaron en la gran final del festival.

  El retorno será siempre el código ISO del país. Tres ejemplos de búsqueda son los siguientes:

```
SELECT code FROM countries WHERE name LIKE '%Alcobendas%' OR name LIKE
'%Madrid%' OR name LIKE '%Spain%' AND EXISTS (SELECT * FROM voters WHERE
voters.code = countries.code)

SELECT country FROM cities WHERE name LIKE '%Madrid%' OR name LIKE '%London%'
AND EXISTS (SELECT * FROM voters WHERE voters.code = cities.country)

SELECT country FROM regions WHERE name LIKE '%En la luna%' AND EXISTS
(SELECT * FROM voters WHERE voters.code = regions.country)
```

  Si no existiesen resultados para la búsqueda realizada, como sucede en el tercer ejemplo, el retorno pasa a ser NULL.

3.5. Indexación de datos

Una vez preprocesados todos los mensajes extraídos de Twitter, disponemos de un directorio con todos los datos guardados en ficheros. El volumen de información que manejamos es inmenso, por lo que el siguiente paso será la organización del contenido de los ficheros en una estructura que permita acceder a cualquier dato o realizar búsquedas de la forma más eficiente posible.

Las bases de datos relacionales son ideales para organizar un conjunto de datos estructuradamente, pero el uso de las mismas en nuestro proyecto tiene algunas desventajas. Por

un lado, las búsquedas son muy costosas, debido a la gran cantidad de información almacenada. Además, analizar cada campo término a término es una tarea laboriosa, teniendo en cuenta que no se manipulan de forma manejable los bloques de texto.

Como alternativa a este tipo de estructura, un índice de datos es una buena opción a tener en cuenta. Los problemas presentes en las bases de datos relacionales, ya mencionados, no aparecen en el uso de índices de datos, puesto que los índices optimizan la tarea de búsqueda y análisis de textos a nivel de término. Por contra, el proceso de indexación de datos es muy costoso. Se describen, en esta sección, los procedimientos de indexación de datos y posterior acceso a ellos.

3.5.1. Construcción del índice

Para crear el índice de datos donde se va a alojar el contenido de nuestros ficheros, se utilizará la herramienta Apache Lucene, una API de código abierto originalmente implementada en Java. Los documentos a indexar por dicha herramienta serán los ficheros generados en la fase de preprocesado, o lo que es lo mismo, se indexarán los tweets extraídos y preprocesados. Únicamente se indexarán los tweets que se consigan geolocalizar, desechándose el resto del conjunto.

Cada documento tendrá una serie de campos, coincidiendo en este caso con los campos del tweet indexado. Además, Lucene dispone de un analizador que realiza una serie de operaciones sobre los campos del documento que nosotros indiquemos. Estas operaciones son principalmente tokenización, filtrado de stopwords, normalización y stemming. Aplicaremos este analizador sobre el campo que contiene el texto del tweet, y de esta forma las búsquedas de términos que hagamos posteriormente en el índice se realizarán sobre este campo.

Cuando el índice haya sido creado, con todos los documentos indexados, podrá ser guardado en un directorio del disco duro para su posterior acceso.

3.6. Generación de rankings

Después de indexar todos los datos extraídos de Twitter y geolocalizados, vamos a organizarlos en rankings. Para ello es necesario realizar búsquedas en el índice para recuperar toda la información, y posteriormente se deberán ordenar los resultados de esas búsquedas.

3.6.1. Conteo del número de menciones acerca de los países

En primer lugar, hemos contado cuántas menciones sobre cada país aparecen entre todos tweets, y además de calcular el total hemos contado cuántas de esas menciones realiza cada país.

Los pasos que hemos seguido para hacer este conteo son los siguientes:

1. Cargamos en memoria el índice creado anteriormente, accediendo al directorio donde fue almacenado en disco en el momento en que se creó.

2. Creamos una lista e insertamos en ella todos los países que actúan en la gala de Eurovisión, escritos todos en inglés por ser el idioma oficial del festival. Los términos de esta lista son los que vamos a buscar en los tweets indexados, para detectar las menciones realizadas acerca de cada uno.

3. Recorremos la lista de países creada en el paso 2, buscando en el índice cada uno de los elementos. Al buscar cada país en el índice, contamos cuántos documentos son relevantes, es decir, cuántos tweets contienen en su campo de texto el país buscado. Esta cantidad será el número de menciones que se han hecho sobre ese país, pero para localizar el país del que procede cada mención debemos recorrer la lista de documentos devueltos en la búsqueda. En el campo place_country_code aparece el código del país que se registró al geolocalizar el tweet, por lo que una forma sencilla de clasificar las menciones por país es rellenar un mapa cuya clave sean los códigos de país y cuyo valor sea el número de veces que se detecta cada código.

El resultado de este conteo será un mapa cuya clave contiene cada país buscado, y cuyo valor contiene el mapa generado en el paso 3.

3.6.2. Cálculo del número de menciones realizadas por cada país

El siguiente paso, ha sido exportar a un fichero de texto el conteo realizado. El fichero, de tipo CSV, contiene el número de menciones que un país ha dado a otro, recogiéndose menciones de todos los países votantes en Eurovisión a todos los países cantantes. Cada línea del fichero contiene un país que realiza menciones, el país al que menciona, y el número de menciones. Se muestra un fragmento del fichero a continuación:

```
From;To;Mentions
Albania;Austria;11
Albania;Poland;11
Albania;Greece;12
Albania;Belarus;9
Albania;United Kingdom;1
Albania;Switzerland;7
Albania;Romania;10
Albania;Finland;8
Albania;Slovenia;10
```

3.6.3. Implementación de clases para el tratamiento de datos en el análisis de los resultados

Antes de comenzar con el análisis de resultados, se han implementado tres sencillas clases java para facilitar el tratamiento de los resultados y el cálculo de las métricas. Perseguimos con esto automatizar lo máximo posible la clasificación por países y la ordenación de elementos en los rankings que estamos generando.

Disponemos de cuatro conjuntos de información en forma de ficheros CSV. Una tupla de cualquiera de estos cuatro ficheros tiene tres datos: país votante, país votado, y puntuación o número de menciones. Se describen a continuación los conjuntos de datos.

3.6.4. Conjuntos de datos

El primer conjunto de datos contiene la cantidad de menciones a países detectadas en los tweets extraídos. Recoge el número de veces que un país menciona a otro, para todos los países participantes.

El resto de conjuntos, han sido obtenidos de la web de Eurovisión[2]. El segundo conjunto de datos contiene las puntuaciones procedentes del televoto. Estas puntuaciones aparecen en forma de posición en ranking, de forma que un país cuyo puesto sea el número 1 entre las votaciones de otro país, tendrá la máxima puntuación. El tercero contiene las puntuaciones procedentes del jurado, y su formato es el mismo que el del fichero anterior. Por último, el cuarto conjunto contiene un ranking que combina las puntuaciones del jurado con las del televoto, y su formato es el mismo que el de los dos ficheros anteriores.

3.6.5. Descripción de las clases

La primera clase, llamada Points, contiene tres atributos: país votante, país votado, y puntuación o número de menciones. Contiene métodos de acceso a los atributos — getters y setters — y sobreescribe el método equals, siendo dos objetos iguales si sus atributos de país votante y país votado contienen lo mismo en ambos. Esto nos facilitará la tarea de comparar la puntuación que un país da a otro con el número de menciones que hace el mismo país votante sobre el mismo país votado.

La segunda clase, llamada Mention, representa un dato entre los resultados de las menciones. Extiende de la clase Points, es implementación de la interfaz Comparable, y sobreescribe el método compareTo para facilitar la ordenación de menciones. De este modo, al ordenar una lista de objetos Mention, el primer elemento será el que más menciones tenga.

La tercera clase, llamada RankingPlace, representa un dato entre los resultados de las puntuaciones, ya sean del jurado, televoto o combinación de ambas. Al igual que la clase Mention, también es extiende de Points, implementa la interfaz Comparable, y sobreescribe el método compareTo. El comportamiento del método compareTo en este caso será inverso al del método compareTo sobreescrito en Mention; al ordenar una lista de objetos de tipo RankingPlace, el valor de la puntuación del primer elemento será el menor de todos, ya que el valor 1 representa al país con más puntos.

[2] http://www.eurovision.tv/page/results

4 Experimentación y análisis

Una vez generado el ranking por países, podemos proceder a su análisis y experimentación. El objetivo del análisis del ranking por menciones es validar si estos datos están relacionados con las votaciones que posteriormente emiten los países durante el festival, o lo que es lo mismo, validar cuán efectiva es nuestra aplicación. A continuación, se aplicarán algunas métricas como la precisión o la distancia Euclidiana, para comprobar si la tendencia que se puede sacar de las opiniones de los usuarios posteriormente se ven reflejadas en las votaciones emitidas por los países. Finalmente, a partir de los resultados del ranking generado, se simularán las puntuaciones del festival.

4.1. Análisis de resultados

Para analizar los resultados obtenidos en este trabajo, se van a comparar los datos del ranking generado con las puntuaciones oficiales de Eurovisión. Las dos métricas que vamos a utilizar son la precisión y la distancia Euclidiana, que toman un valor entre 0 y 1. Cuanto más se acerque a 1 la precisión, más acertados serán los resultados, y cuando más se acerque a 1 la distancia Euclidiana, más cercano será nuestro ranking de menciones al real. Originalmente, la distancia Euclidiana puede tomar cualquier valor positivo, pero en este trabajo ha sido normalizada. Disponemos, para realizar el análisis, de cuatro conjuntos de datos organizados ya en rankings.

4.1.1. Características del conjunto de información extraída de Twitter

Se consiguió realizar una extracción de 155.422 tweets, obtenidos al hacer la búsqueda del hashtag #Eurovisión. Posteriormente, un 78 % de los tweets fueron geolocalizados con éxito, concretamente una cantidad de 121.440 mensajes. Todos estos tweets fueron escritos en los días 7, 8 y 9 de mayo, y el día 10 de mayo hasta horas antes del festival, celebrándose éste ese mismo día por la noche.

4.1.2. Precisión de los resultados

El primer cálculo que se ha hecho en el análisis de los resultados ha sido la precisión. Dicho valor mide el acierto al predecir si un país ha sido votado por otro, independientemente de la puntuación que emita. La fórmula de la precisión queda recogida en la Ecuación 4.1

[10], donde *TP (True Positive)* son los países considerados correctamente como votados y *FP (False Positive)* son los países erróneamente considerados como votados.

$$Prec = \frac{TP}{TP + FP} \tag{4.1}$$

Un caso específico en el cálculo de la precisión es *P@k (Precision at k)*, que consiste en calcular la precisión sobre un ranking de tamaño k.

Esta métrica se aplica sobre cada país votante, y el procedimiento a seguir es el siguiente:

1. Tomamos el ranking de puntuaciones otorgadas por cada país, y también el ranking de menciones que realiza en Twitter el mismo país, ambos ordenados de mayor a menor.

2. Elegimos un valor para k, que será el número de elementos — países — que seleccionemos en el ranking, y sobre los que calcular la tasa.

3. Contamos el número de países que se encuentran tanto en el ranking de los k países más mencionados como en el ranking de los k países más puntuados. La precisión será el resultado del cociente del número de coincidencias entre k.

El cálculo de la precisión se ha aplicado por separado sobre los tres rankings — voto del jurado, televoto, y combinación de los dos anteriores — y sobre cada país votante. El valor de k elegido ha sido $k = 10$, ya que en la gala de Eurovisión cada país puntúa a los diez países mejor posicionados en su ranking de votaciones.

Los resultados del cálculo de la precisión al comparar nuestro ranking de menciones con las puntuaciones oficiales se recogen en la Tabla 4.1, mostrando los valores hallados para cada país votante. Albania y San Marino no tienen acceso al sistema del televoto, y Georgia no dispone de un jurado que realice puntuaciones, por lo que para estos tres países solo se ha calculado la métrica de la precisión sobre dos rankings.

Por lo general, observamos que la precisión correspondiente al ranking de televoto supera a la precisión en los otros casos, obteniéndose un 41 % acierto. Esto nos indica que los resultados obtenidos en el ranking de menciones son mucho más fieles a las puntuaciones procedentes del televoto que a las puntuaciones del jurado. Mientras que el televoto manifiesta la opinión de la gente, el voto del jurado pertenece a un grupo de cuatro o cinco personas, profesionales en el mundo de la música, quienes evalúan de manera más técnica las actuaciones. Los tweets, al igual que el televoto, manifiestan también lo que la gente opina, lo cual explica que la precisión sea mayor en un caso que en otro. Por otro lado, la precisión correspondiente al ranking de voto combinado, se suele situar entre los valores de esta métrica para los otros dos rankings, ya que el voto combinado resulta del cálculo de la media entre el voto del jurado y el televoto.

Como conclusiones acerca de estos resultados, podemos decir que uno de los factores que han influido en ellos ha sido el idioma empleado en el proceso de minería de datos. La base de datos World está escrita en inglés, lo cuál afecta directamente a la geolocalización de tweets. Todos los nombres que tengan traducción al inglés — por ejemplo, España

País	**Televoto**	**Voto del jurado**	**Voto combinado**
Belgium	0.5	0.3	0.3
Denmark	0.5	0.3	0.3
Georgia	0.5	-	0.5
Hungary	0.5	0.1	0.4
Iceland	0.5	0.4	0.5
Ireland	0.5	0.5	0.5
Italy	0.5	0.5	0.7
Norway	*0.5*	*0.6*	*0.4*
Malta	0.5	0.4	0.5
United Kingdom	0.5	0.5	0.5
Armenia	0.4	0.2	0.3
Austria	0.4	0.4	0.4
Azerbaijan	0.4	0.4	0.5
Belarus	0.4	0.3	0.3
France	0.4	0.3	0.3
Germany	0.4	0.3	0.4
Greece	*0.4*	*0.5*	*0.4*
Latvia	0.4	0.4	0.4
Lithuania	0.4	0.2	0.3
Moldova	0.4	0.3	0.4
Montenegro	0.4	0.4	0.5
Poland	0.4	0.4	0.3
Slovenia	0.4	0.3	0.4
Spain	0.4	0.4	0.3
Sweden	0.4	0.3	0.4
Switzerland	0.4	0.2	0.3
Ukraine	0.4	0.4	0.3
Estonia	0.3	0.3	0.3
Finland	0.3	0.3	0.3
Israel	0.3	0.3	0.3
Portugal	0.3	0.3	0.4
Romania	0.3	0.3	0.2
Russia	0.3	0.3	0.3
The Netherlands	0.3	0.2	0.3
Albania	-	0.4	0.4
San Marino	-	0.2	0.2
Valor medio	0.4088	0.3400	0.3750

Tabla 4.1: Resultados del cálculo de la precisión

se traduce como Spain — aparecerán en World escritos en inglés. Por tanto es probable que, al buscar un nombre escrito en otro idioma, no obtengamos resultados. Además, para buscar menciones sobre países en los tweets extraídos, se ha utilizado la lista de países participantes traducida al inglés. Países como Inglaterra o Irlanda tienen una de las tasas de acierto más altas, reforzando nuestra teoría; sin embargo, países como Rusia tienen la

precisión más baja, lo cual tiene sentido porque en Rusia se utiliza incluso otro alfabeto distinto.

Por otra parte, se decidió analizar en primera instancia si la frecuencia de las menciones de un país a otro estaba estrechamente relacionada con las puntuaciones emitidas. Como la precisión no supera ningún caso un valor de 0.5, sería conveniente realizar un análisis de sentimientos sobre la información extraída. No solo hemos considerado los tweets positivos acerca de un país, sino que también hemos considerado los tweets negativos, por lo que tras realizar un análisis de sentimientos deberíamos filtrar solo los que contengan opiniones positivas acerca de los países a los que se menciona.

4.1.3. Distancia Euclidiana

La precisión de los resultados mide la tasa de aciertos entre las 10 primeras votaciones de un país, pero no es una medida fina para evaluar los resultados. Esta métrica no calcula la diferencia entre la posición de un país en nuestro ranking de menciones y la posición del mismo país en el ranking de votaciones, de manera que si un país ocupa el primer lugar entre las votaciones y el décimo en nuestro ranking, se convierte en un caso de acierto entre los resultados. Para afinar el análisis de los resultados, debemos calcular la distancia Euclidiana.

Esta métrica mide la distancia entre dos puntos, en nuestro caso países, pudiendo ser representadas ambos como vectores. El número de elementos define el número de dimensiones, en este caso 26 por ser el número de países a los que se les pueden dar votos, y la frecuencia de los términos dará valor a las coordenadas a lo largo de esas dimensiones, traduciéndose este valor al número de votos. La distancia se calcula como la raíz cuadrada de la suma de los cuadrados de las diferencias en una posición a lo largo de cada dimensión [11]. Coloquialmente, esta sería la distancia que podríamos medir con una regla, y su definición se deduce a partir del teorema de Pitágoras. La fórmula del cálculo de la distancia Euclidiana es la que se muestra en la Ecuación 4.2.

$$d_e(X,Y) = \sqrt{\sum_{i=0}^{n}(X_i - Y_i)^2} \tag{4.2}$$

Al igual que sucede en el procedimiento de cálculo de la precisión, se tomarán el ranking de puntuaciones y el de menciones ordenados de menor a mayor, para cada país votante. Seguidamente, y tomando de nuevo $k = 10$, sustituimos la cantidad de puntos y menciones de cada país en el ranking: 10 puntos o menciones para el primero, 9 para el segundo, y así sucesivamente hasta 0. A partir del país en la posición número 11, el valor puntos o menciones será 0.

Tras calcular la distancia Euclidiana, donde el primer vector son los valores de 0 a 10 correspondientes al ranking de menciones y el segundo vector son las puntuaciones de 0 a 10 correspondientes a un ranking de votaciones, vamos a normalizar los resultados. Para ello debe calcularse la distancia Euclidiana máxima, partiendo del caso peor, recogido en la Tabla 4.2. Los valores numéricos de la tabla representan la puntuación en el ranking de menciones o votaciones de un mismo país a los 26 países que participan. Aplicando la

fórmula de la distancia Euclidiana (Ecuación 4.2) sobre este caso obtenemos la distancia Euclidiana máxima, con un valor de 27.74.

Menciones	10	9	8	7	6	5	4	3	2	1	...	0	0	0	0	0	0	0	0	0	0
Votos	0	0	0	0	0	0	0	0	0	0	...	1	2	3	4	5	6	7	8	9	10

Tabla 4.2: Resultados del cálculo de la distancia Euclidiana

Una vez obtenida la distancia Euclidiana máxima, el cálculo de la distancia Euclidiana normalizada es el que resulta de la Ecuación 4.3.

$$d_{Norm}(X,Y) = 1 - \frac{d_e(X,Y)}{d_{Max}(X,Y)} \tag{4.3}$$

Los resultados del cálculo de esta métrica se recogen en la Tabla 4.3. Al igual que sucede en el cálculo de la precisión de los resultados, generalmente la distancia Euclidiana normalizada es mayor — siendo menor la distancia Euclidiana original — si se calcula sobre el ranking de televotos que si se calcula sobre el ranking de votaciones del jurado. En este caso, la distancia Euclidiana normalizada alcanza como mucho un valor de 0.45 si se aplica sobre el ranking del televoto, y de media vale aproximadamente 0.3. También vuelve a haber dos casos extraños: Azerbaijan e Israel tienen una distancia más favorable en el ranking del jurado que en el de menciones.

Esto quiere decir que las puntuaciones en el ranking de menciones suelen ser lejanas a las puntuaciones en los rankings de votaciones. En otras palabras, los países con puntuaciones muy altas en un ranking son países con puntuaciones muy bajas en otro ranking, y viceversa. Como conclusión obtenemos que esta aplicación aún no puede realizar predicción alguna sobre las puntuaciones del festival, puesto que la distancia Euclidiana no llega al 0.5. Una opción para detectar errores en la clasificación en el ranking, sería variar el valor de k, tomando nota de la distancia Euclidiana en cada caso.

Como complemento a la tabla de los valores de las distancias, se ha creado un gráfico de barras (Figura 4.1) que muestra visualmente las distancias Euclidianas para el caso del ranking de los televotos. El gráfico ha sido generado a partir de los valores de las distancias normalizadas, pero cuanto mayor sea la barra mayor será la distancia sin normalizar. La línea vertical en el gráfico representa la media.

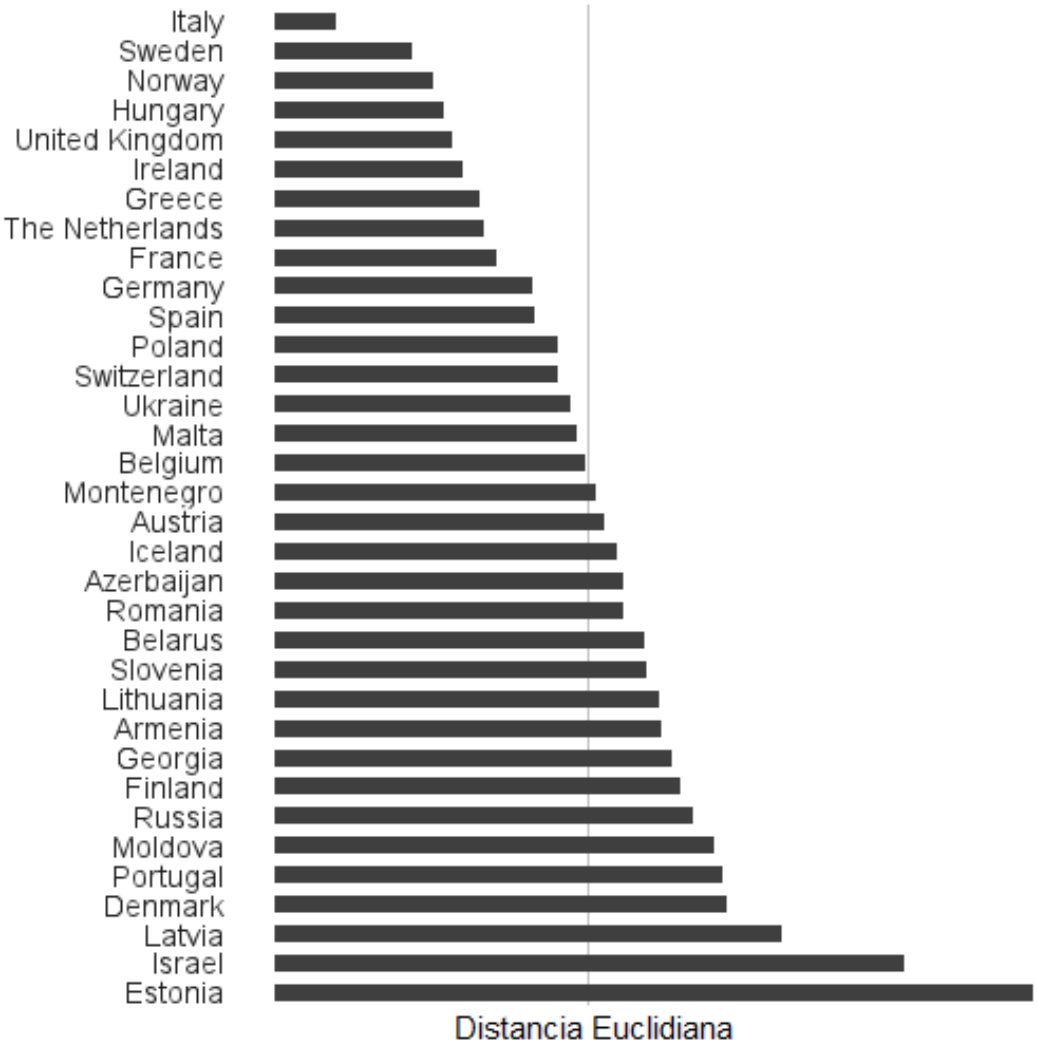

Figura 4.1: Representación visual de la distancia Euclidiana para todos los países

País	**Televoto**	**Voto del jurado**	**Voto combinado**
Italy	0.451	0.363	0.423
Sweden	0.398	0.284	0.313
Norway	0.385	0.342	0.350
Hungary	0.378	0.084	0.277
United Kingdom	0.374	0.227	0.246
Ireland	0.368	0.260	0.230
Greece	0.357	0.301	0.265
The Netherlands	0.354	0.233	0.243
France	0.347	0.183	0.232
Germany	0.328	0.108	0.202
Spain	0.327	0.280	0.266
Poland	0.314	0.251	0.201
Switzerland	0.314	0.296	0.253
Ukraine	0.309	0.261	0.269
Malta	0.305	0.206	0.286
Belgium	0.301	0.177	0.207
Montenegro	0.296	0.077	0.172
Austria	0.292	0.180	0.181
Iceland	0.286	0.255	0.291
Azerbaijan	*0.283*	*0.316*	*0.276*
Romania	0.283	0.146	0.193
Belarus	0.274	0.158	0.225
Slovenia	0.272	0.224	0.236
Lithuania	0.267	0.170	0.173
Armenia	0.266	0.098	0.192
Georgia	0.261	-	0.261
Finland	0.258	0.249	0.254
Russia	0.253	0.227	0.249
Moldova	0.244	0.181	0.241
Portugal	0.241	0.173	0.263
Denmark	0.239	0.217	0.202
Latvia	0.219	0.063	0.152
Israel	*0.180*	*0.246*	*0.175*
Estonia	0.146	0.071	0.138
Albania	-	0.120	0.120
San Marino	-	0.088	0.088
Valor medio	0.299	0.203	0.232

Tabla 4.3: Resultados del cálculo de la distancia Euclidiana

4.2. Predicción de resultados

Procesando los datos de nuestro ranking de menciones generado, se ha procedido a predecir un ranking con las puntuaciones definitivas que obtendría cada país en la gala del festival. Para realizar la predicción de puntuaciones, el procedimiento ha sido exactamente el mismo que se sigue en Eurovisión: cada país dará 12 puntos a su país más votado, al segundo más votado le dará 10 puntos, al tercero 8 puntos, y al resto de 7 a 1, votando únicamente a los 10 mejores.

Considerando el país más mencionado en otro país, como si fuese el más votado, se le sumarán 12 puntos, al siguiente más mencionado se le sumarán 10 puntos, y así sucesivamente siguiendo el mismo sistema, como ya se ha comentado. En la Tabla 4.4 se muestran las puntuaciones de nuestra predicción frente a las puntuaciones reales, y el número de posición en el ranking de la predicción frente al número de posición en el ranking real.

País	**Puntos (predicción)**	**Puntos (real)**	**Puesto (predicción)**	**Puesto (real)**
Austria	402	290	1	1
Greece	297	35	2	20
Poland	261	62	3	14
Finland	234	72	4	11
Belarus	193	43	5	16
Romania	175	72	6	12
Switzerland	165	64	7	13
Slovenia	102	9	8	25
Malta	69	32	9	23
Norway	59	88	10	8
U. Kingdom	42	40	11	17
Italy	31	33	12	22
France	20	2	13	26
Azerbaijan	12	33	14	21
Netherlands	12	238	15	2
Spain	10	74	16	10
Armenia	10	174	17	4
Iceland	10	58	18	15
Hungary	8	143	19	5
Sweden	8	218	20	3
Ukraine	7	113	21	6
Denmark	7	74	22	9
San Marino	6	14	23	24
Montenegro	2	37	24	19
Russia	2	89	25	7
Germany	2	39	26	18

Tabla 4.4: Predicción de puntuaciones

Para comparar qué países están situados en las posiciones más importantes de cada ranking, de manera más clara, se muestran en la Tabla 4.5 los 5 primeros países del ranking

de nuestra predicción y los 5 primeros países del ranking real.

Top 5 predicho	**Top 5 real**
Austria	Austria
Greece	The Netherlands
Poland	Sweden
Finland	Armenia
Belarus	Hungary

Tabla 4.5: Países del *Top* 5 en los rankings finales

Observamos que lo que el sistema ha predicho acerca del ganador se cumple, existiendo en la predicción una amplia diferencia entre las puntuaciones del primer y segundo puesto. Pero comparamos las listas del *Top* 5 de la predicción y de las puntuaciones reales y no hay más coincidencias.

Concluimos en que, para predecir el ganador del festival, podemos hacer un estudio en el que únicamente se tenga en cuenta la frecuencia de las menciones en lo tweets. Pero, para afinar las predicciones en el resto del ranking, deberán realizarse una serie de mejoras ya propuestas en la Sección 4.1.2, donde se calcula la precisión, y analizarse la distancia en función del valor de k para detectar cuáles son los datos que más nos perjudiquen.

5 Conclusiones y trabajo futuro

En este trabajo, se ha desarrollado una aplicación para la detección de tendencias en Twitter, basada en técnicas de minería de datos. Estas técnicas consisten en el proceso del descubrimiento de patrones, tendencias y nuevas relaciones al examinar grandes cantidades de datos, obteniéndose de manera automática nuevos conocimientos.

Para realizar un estudio de forma acotada, hemos elegido Eurovisión como tema principal, ya que gracias a su sistema de televoto podemos conseguir información muy abundante y valiosa, si la explotamos de manera eficaz. Las fases del proceso de minería de datos por las que este proyecto ha pasado, han sido las siguientes:

1. Filtrado de datos, en este caso extracción de tweets. Se han recopilado en total una cifra de 155.422 mensajes, entre los días 7 y 10 de mayo, ambos inclusive.
2. Preprocesado de la información extraída. Se ha dado un formato especial a los tweets extraídos en la fase anterior, geolocalizando los que no estuviesen geolocalizados por defecto. Tras esta fase, se han obtenido un total de 121.440 tweets geolocalizados.
3. Indexación de los datos preprocesados. Un índice nos permite realizar búsquedas de manera eficiente sobre los datos, lo cual ha sido determinante para su selección como estructura de almacenamiento de datos.
4. Generación de un ranking de menciones por países, recogiendo el número de menciones realizadas de un país a otro, y ordenado de mayor a menor en el número de menciones. Se han generado también tres rankings que recogen las puntuaciones del televoto, las del voto del jurado y las del voto combinado.
5. Análisis de los resultados recogidos en el ranking, aplicándose dos métricas distintas: precisión de los resultados y distancia Euclidiana.

Una vez realizado el análisis de resultados, se observa que la precisión no supera el 50 % en ningún caso, y que la distancia Euclidiana toma un valor bastante alto. Concluimos con esto en que, al tomar como objeto de estudio únicamente la frecuencia de las menciones es imposible predecir resultados, ya que se tienen en cuenta tanto las buenas como las malas opiniones de los usuarios. Además, la aplicación solo entiende un idioma, el inglés, limitando así la explotación de datos.

Por último, se ha realizado una predicción acerca de la puntuación que obtendría cada país en la gala de Eurovisión, y efectivamente los resultados reales han sido impredecibles, aunque la aplicación ha acertado con éxito al ganador del festival.

Quedan pendientes, como trabajo futuro, algunas tareas para mejorar este proyecto:

- Para aumentar la eficacia de nuestra aplicación, podría implementarse un clasificador con el fin de realizar un análisis de sentimientos, permitiéndonos detectar qué tweets son positivos cuáles son negativos en cuanto a las opiniones de los usuarios.
- También será necesario que el sistema sea multi-idioma, utilizando otras bases de datos o ampliando la que ya tenemos para una geolocalización más precisa, y traduciendo a otros idiomas los nombres de los países a buscar para la detección de las menciones.
- Por supuesto, seguir extrayendo y recopilando tweets. Cuanto mayor sea el volúmen de datos, mayor será la información que obtengamos.

Glosario

- **Clustering**: Técnica de aprendizaje no supervisada que consiste en la búsqueda de agrupamientos dentro de un conjunto de datos.
- **Geolocalización**: Localización de un elemento vía GPS, aportando información acerca de la situación geográfica del mismo.
- **Hashtag**: Término precedido del símbolo '#' que sirve para etiquetar un mensaje en Twitter.
- **Microblogging**: Tipo de servicio que consiste en el envío y publicación de mensajes breves.
- **Normalización** (de una palabra): Eliminación acentos y mayúsculas en una palabra.
- **Retwittear**: Compartir, en nuestro propio timeline, el tweet que ha publicado otro usuario.
- **Spam**: Mensajes de correo no deseado, aunque puede difundirse a través de otros medios de comunicación, como microblogging. Suelen ser mensajes de tipo publicitario.
- **Steming**: Reducción de las derivaciones e inflexiones en una palabra, como su número, género, conjugación, etc. Por ejemplo, *am* se convierte en *be*, o *cars* se convierte en *car*.
- **Stopwords**: Partículas funcionales con poco significado propio. Son palabras de este tipo los artículos, las preposiciones, las conjunciones o los adjetivos demostrativos.
- **Timeline**: Histórico de los mensajes de un usuario en Twitter.
- **Tokenización**: Separación de un texto en palabras.

Bibliografía

[1] César Pérez López. *Minería de datos: técnicas y herramientas.* Editorial Paraninfo, 2007.

[2] Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188, 2009.

[3] Kathy Lee, Ankit Agrawal, and Alok Choudhary. Real-time disease surveillance using twitter data: demonstration on flu and cancer. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1474–1477. ACM, 2013.

[4] Tim Highfield, Stephen Harrington, and Axel Bruns. Twitter as a technology for audiencing and fandom: The# eurovision phenomenon. *Information, Communication & Society*, 16(3):315–339, 2013.

[5] J.K. O'Connor, Australian Broadcasting Corporation, and S. Stockselius. *The Eurovision Song Contest - 50 Years: The Official History.* ABC Books, 2005.

[6] Gad Yair. 'unite unite europe' the political and cultural structures of europe as reflected in the eurovision song contest. *Social Networks*, 17(2):147 – 161, 1995.

[7] Gema Bello Orgaz, Raúl Cajias, and David Camacho. A study on the impact of crowd-based voting schemes in the 'eurovision' european contest. In Rajendra Akerkar, editor, *WIMS*, page 25. ACM, 2011.

[8] Daniel Fenn, Omer Suleman, Janet Efstathiou, and Neil F. Johnson. How does europe make its mind up? connections, cliques, and compatibility between countries in the eurovision song contest. *Physica A: Statistical Mechanics and its Applications*, 360(2):576–598, 2006.

[9] European Broadcasting Union 2004-2014. Eurovisión Scoreboards in 2014 http://www.eurovision.tv/page/results.

[10] Gerard Salton and Michael J. McGill. *Introduction to modern information retrieval.* McGraw-Hill, New York, 1983.

[11] Donald G. Bailey. An efficient euclidean distance transform. In *Proceedings of the 10th international conference on Combinatorial Image Analysis*, IWCIA'04, pages 394–408, Berlin, Heidelberg, 2004. Springer-Verlag.